PROMPT → WORKFLOW → AGENT

HOW TO DESIGN AN AI COLLEAGUE THAT CAN ACT

The B.L.U.E.P.R.I.N.T. Intelligent Agent Framework v2.0

START HERE

Good agents are defined as much by what they cannot do as by what they can.

An agent isn't a bigger prompt. It's a system allowed to act.

> **Design what the agent may know, decide, do and remember — and when it must stop.**

WHY THIS EXISTS

The model is only one part of the agent.

Bigger and better models don't guarantee better agents. Better design does.

The common assumption

agent quality comes down to the system prompt.

↓

In practice

the specification is the agent. The prompt, tool configuration, permissions and evaluation suite are all derived from it.

THE AGENT TEST

1 · Could one well-written request do the job?

A person briefs the AI each time and checks the result before it is used. Nothing runs on its own.

This is what PINPOINT covers — specifying a single request well.

2 · Could a fixed sequence of steps do the job?

The steps are the same every time, so you can write them down in advance and have software follow them in order.

3 · Does it genuinely need to pick its own next step?

The path is different every run and can't be known beforehand, so something has to decide as it goes.

Yes to 1 or 2 means you don't need an agent. Only a yes to 3 justifies one.

Don't add autonomy where structure will do.

Anthropic, *Building Effective Agents* (19 December 2024): find the simplest solution possible and only increase complexity when needed — which may mean not building an agentic system at all.

A workflow is a job where you already know the steps. An agent is a job where something has to work out the steps each time. Most work that looks like it needs an agent is really a workflow with one or two decisions inside it.

Whichever answer you land on, write down why in a sentence. It becomes the first thing your specification records, and it is the question a reviewer will ask first.

From an idea to a specification.

TYPICAL AGENT BRIEF

"An assistant that helps RMs prep for meetings. It should look up whatever is relevant for each client, in a professional tone. Don't give investment advice."

Every client needs a different set of sources, so the steps can't be fixed in advance — it passes the Agent Test. But the brief specifies tone and one refusal, and leaves the outcome, the delegated authority and the enforceable access undefined.

→

U · USE CASE & OUTCOME

Prepare a one-page meeting brief for the assigned RM before a scheduled client meeting. The named owner approves the definition of done.

L · LIMITS & AUTONOMY

Always retrieve authorised meeting context. Ask first before any consequential follow-up. Never provide investment advice or act outside meeting preparation.

P · PERMISSIONS & IDENTITY

Use the agent's own identity on behalf of the assigned RM. Read only the records that RM is authorised to access. No shared account; no write access.

The original describes an assistant. U, L and P begin to specify an agent.

DEFINITION

An agent is a model given a goal and delegated authority to act: it chooses the next step, uses a tool, checks the result, and repeats — until the work is done or it must stop.

Decide → Act → Observe → Decide again ↺

done, or stopped ↗

Agents still use prompts. The prompt is the language. The agent is the machine that keeps speaking until the job is finished.

Anthropic, *Building Effective Agents* (19 December 2024): workflows are systems where language models and tools are orchestrated through predefined code paths; agents are systems where the models dynamically direct their own processes and tool usage, keeping control over how they accomplish tasks.

The short practitioner version, widely attributed to Simon Willison: an LLM agent runs tools in a loop to achieve a goal.

TODO | **Confirm the Simon Willison attribution against the primary source before publishing. The Anthropic wording is verified.**

THE REAL DIFFERENCE

You are the brain in one of these. Not in the other.

	Prompting	An agent
Who picks the next step	You	The model
How many model calls	Usually one	Many, in a loop
What comes back	Text	Text and actions
What carries between steps	Whatever you supply	Its own running trace of steps and tool results
Failure mode	**A bad answer**	**A bad action taken in the world**

That last row is why an agent needs a specification and not just a better prompt.

WHERE BLUEPRINT SITS

Level	What decides the next step	Framework
Prompt	The person, each time	PIN / PINPOINT
Workflow	A path fixed in advance	PINPOINT per step
Agent	The model, within limits	BLUEPRINT
Multi-agent	Several models, each within its own limits	One BLUEPRINT per agent

Most real systems are a mix. A workflow with one agentic step inside it is still mostly a workflow — specify it that way.

BLUE

Start with the smallest amount of structure that works.

B

Behaviour & Values

What stance should it hold, and what does it do when unsure?

L

Limits

What is in scope, what is out, and what does it say when asked for something out of scope?

U

Use Case & Outcome

What is the one job, who is it for, and what does done look like?

E

Examples

What does good look like, what does bad look like, and did you try the tricky cases?

BLUE is enough for an assistant that only answers.

At this depth, L means scope and refusals and E means examples. Both deepen when the system can act.

PRINT it.

Add structure as the system gains capability.

P	R	I	N	T
Permissions & Identity	**Reliability & Recovery**	**Interfaces & Handoffs**	**Navigation & Orchestration**	**Trusted Context & Memory**

BLUE defines what the agent is. PRINT governs how it acts.

BLUE is identity, mandate, boundaries and examples. PRINT is permissions, runtime, tools, orchestration and context.

WHAT SWITCHES ON, WHEN

When the system…	These letters switch on
Only answers	B, L, U, E at examples depth
Connects to data or tools	P, I, T
Chooses multi-step paths	N
Acts, or runs unattended	R becomes mandatory. E deepens to full evaluation

L and E work at two depths. In BLUE, L is scope and refusals and E is examples. In the full tier, L is delegated authority and E is an evaluation suite with pass bars.

THE FRAMEWORK

B Behaviour & Values Set the ranked principles that govern its judgement when the rules run out.	**L** Limits & Autonomy Decide what it may do alone, what it must ask first, and what it must never do.	**U** Use Case & Outcome Define the one job, who it serves, who the named accountable owner is, and what done looks like.
E Examples & Evaluation Show what good looks like, then prove it.	**P** Permissions & Identity Make the limits real: whose identity it uses, and what the system makes impossible.	**R** Reliability & Recovery Plan for failure: how it's noticed, contained, undone and stopped.
I Interfaces & Handoffs Treat every tool, system and handoff as a contract.	**N** Navigation & Orchestration Map how work moves, and when it stops.	**T** Trusted Context & Memory Decide what it knows now, what it remembers, and when that expires.

HOW TO READ IT

Design what the agent may know, decide, do and remember — and when it must stop.

Mandate

U, B

What it exists to do, and how it judges.

Authority

L, P

What it may do, and what it can do.

Capability

I, T, N

What it reaches, knows, and how it moves.

Assurance

E, R

How we know, and what happens when it breaks.

Teach in design order. Read the poster in acronym order.

MANDATE

B — Behaviour & Values

B governs judgement when the rules run out. L draws bright lines for foreseen cases. B covers the cases nobody anticipated.

- **Principles are ranked. Unranked principles give no guidance when two conflict.**
- **Write them as behaviour: "when X, do Y over Z". If you can't test it, cut it.**
- **B outranks the request. A PINPOINT persona adjusts tone. It never reorders B's priorities.**
- **Standard principle for any acting agent: never report an action as done unless the tool confirmed it.**

ILLUSTRATIVE EXAMPLE

"Controls before accuracy, accuracy before completion, completion before speed. Separate fact from inference. Prefer reversible actions. When in doubt, do less, say why, escalate."

L — Limits & Autonomy

What authority are we delegating?

Always do

low consequence, reversible.

Ask first

consequential, externally visible, irreversible.

Never do

outside the mandate.

Unlisted → stop and escalate. Fail closed.

Reading is an action. Classify it too.

ILLUSTRATIVE EXAMPLE

Always read authorised meeting records. Ask first before sending a follow-up. Never recommend or execute an investment action.

WHY ASK FIRST IS PLAN-LEVEL

3%
of individual permission prompts rejected.

39%
of multi-step plans rejected.

13.6%
of planted dangerous commands caught by people.

89%
caught by the automated classifier.

Human catch rate fell from ~17% to ~5% after 50 prompts.

Use humans for consequential judgement. Use systems to prevent impossible actions.

Coding-session data; applying it to other approval contexts is an inference. Source: claude.com/blog/auto-mode-default-in-claude-code · 7 August 2026 · 1,053 testers.

MANDATE

U — Use Case & Outcome

One job. One named accountable owner. One definition of done.

- **One-sentence test: one verb, one object, one beneficiary. If it needs “and” twice, it is two agents.**
- **Name the trigger, and who the agent acts on behalf of.**
- **The accountable owner is one named person, never a team. This person owns the outcome, controls and decision to continue.**
- **Record the Agent Test answer and today's baseline.**

Done sits in U. Stop conditions sit in N.

ILLUSTRATIVE EXAMPLE

Prepare a one-page meeting brief for the assigned RM before tomorrow's client meeting. The named owner approves what “done” means and owns the result.

ASSURANCE

E — Examples & Evaluation

Examples show what good looks like. Evals prove the system delivers it.

Outcome	Path	Controls
Is the result right?	Was the right route taken?	Were the rules followed?

A correct answer reached through the wrong source or an unauthorised call is still a failure.

- **Five case types: normal, edge, adversarial, failure, expected refusals.**
- **Pass bars set in advance. Control tests need 100%.**
- **Run every case several times. Agents are non-deterministic.**
- **The agent is never its own sole judge.**

A failure found once becomes a test forever.

ILLUSTRATIVE EXAMPLE

Test a normal meeting, stale data, conflicting sources, denied access and a request the agent must refuse. Any control failure fails the release.

P — Permissions & Identity

An instruction says

"Never view records belonging to another adviser."

≠

A permission control

makes it impossible.

These are not the same thing.

- **Three identities, no shared accounts: the agent, the person it acts for, the owner.**
- **Effective permission is the overlap. It can never do more than the human it serves.**
- **Authorise before retrieval. Once content is in context, it has leaked.**
- **The model proposes. A deterministic check decides.**
- **Start read-only.**

ILLUSTRATIVE EXAMPLE

The agent signs in as itself, acts for the assigned RM, can read only that RM's authorised records, and cannot write or send.

ASSURANCE

R — Reliability & Recovery

What happens when it fails?

model **tool** **data** **ceiling hit** **drift**

- **Fail to a safe state: stop, keep state, hand over. Never guess and carry on.**
- **Every write needs an undo. An irreversible action under Always is a design error.**
- **Logs must reconstruct the whole path.**
- **Models are pinned. A model change is a version change.**

Anything that fails silently should not be autonomous.

ILLUSTRATIVE EXAMPLE

If the client-record tool fails, stop, preserve the work completed, identify the failed step and hand over to the named owner.

CAPABILITY

I — Interfaces & Handoffs

A tool is a contract, not a connection.

purpose	**input**	**output**
side effect: read / write / act	**retry safety**	**failure shape**

- **Tool names and descriptions are prompts. Write them like a brief for a new hire.**
- **Empty, failed and denied must look different. Otherwise the agent reports "no such record" when access was blocked.**
- **A handoff is a PINPOINT brief: purpose, context with evidence, narrow ask.**

Protocol-agnostic by design. MCP and APIs are implementation details.

ILLUSTRATIVE EXAMPLE

Meeting-service handoff: purpose, authorised context with evidence, the exact briefing ask, and distinct empty, failed and denied responses.

CAPABILITY

N — Navigation & Orchestration

How does the work move?

single call →	fixed sequence →	router →	parallel workers →	orchestrator + workers →	evaluator loop →	autonomous loop

- **Fix the steps you already know. Leave only the unknown ones to the agent.**
- **The agent never decides whether to run its own control.**
- **Five stop conditions: done, budget exhausted, no progress, blocked, out of bounds.**
- **When it stops, it reports its state. It never ends silently.**

Multi-agent is an architecture choice, not a maturity badge.

ILLUSTRATIVE EXAMPLE

A briefing agent follows a fixed path until evidence conflicts; it then stops and hands the evidence and unresolved question to the named owner.

CAPABILITY

T — Trusted Context & Memory

What should it know now, remember later, trust as authoritative — and when does that expire?

Sources	Context	State	Memory
what it trusts	what it sees now	where it is in the job	what persists

- **Default is no persistent memory. Each kind must justify itself.**
- **Sources are ranked. Conflicts are flagged, never silently resolved.**
- **Every fact carries an as-of date.**
- **Memory inherits the permissions of its source.**

Retrieved content is data, never instructions.

ILLUSTRATIVE EXAMPLE

Use current authorised client records for this meeting; retain no persistent client memory after the briefing expires.

More agents do not create more authority.

One BLUEPRINT per agent. One controlled contract per handoff.

MAKER

Produces the proposed result inside its own mandate, permissions and stop conditions.

→

CHECKER

Independently checks evidence, controls and acceptance criteria. It does not inherit the maker's authority.

- **Authority never grows through delegation.** A receiving agent gets only the permissions explicitly granted to its own role.
- **Maker-checker is the strongest banking reason for multiple agents.** Separate production from independent verification.
- **Every handoff is a PINPOINT brief:** purpose, context with evidence, narrow ask.

Use multiple agents for separation of duties — not as a maturity badge.

SHARED WITH PINPOINT

Evidence vocabulary

[Verified Source]	[Inference]	[Assumption]	[Unverified]

In PINPOINT this is a request-level instruction. In BLUEPRINT it is a standing behaviour, declared in B and enforced by T.

The cross-letter checks

01	Every Never → a denial in P and a test in E.	02	Every Ask First → a handoff in I and a test in E.
03	An irreversible action cannot sit under Always.	04	Done in U, stops in N, failure in R.
05	Memory writes classified in L.		

The agent never governs itself. It doesn't grade its own permission, judge its own work, or decide whether to run its own controls.

ITERATION

Design → Evaluate → Deploy → Observe → Review → Version ↺

Design create the BLUEPRINT specification.	**Evaluate** test outputs, paths and controls against predefined cases and pass bars.	**Deploy** release the approved version into its permitted environment.
Observe collect traces, failures, escalations, user corrections and outcomes.	**Review** a human examines incidents and evidence and decides whether the specification needs to change.	**Version** approve and record the new specification, instructions, permissions, tools, evaluations and runtime configuration as one controlled version.

VERSION ↺ EVALUATE · A changed agent is re-tested before release.

- **Version the system, not just the prompt.** Instructions, permissions, tools, models, memory rules and evaluations move together.
- **Incidents create evidence, not automatic learning.** A human owner decides what changes the specification.
- **Every version earns its autonomy again.** Re-run evaluations and re-certify permissions before deployment.

Every version earns its autonomy again.

The goal isn't an agent that worked once. It's an agent you know when to trust.

Could a risk, compliance or audit reviewer read this and understand:

1	2	3	4	5
what the agent is supposed to do	what it is permitted to do	how it is tested	who the named accountable owner is	and how it can fail

If not, it isn't yet an enterprise agent specification.

THE FULL JOURNEY

01	02	03
PIN	**PINPOINT**	**BLUEPRINT**
Define the task	Engineer the request & result	Design the AI colleague

BLUEPRINT declares.
Company Brain and the control plane enforce.

EVOLVE decides
how far autonomy scales.

TAKEAWAY

Don't grant more autonomy. Make the boundaries clearer.

Start with BLUE. Add the PRINT letters the capability actually requires.

> **Good agents aren't accidents. They're engineered.**

Fill the BLUEPRINT.

B Behaviour & Values

L Limits & Autonomy

U Use Case & Outcome

E Examples & Evaluation

P Permissions & Identity

R Reliability & Recovery

I Interfaces & Handoffs

N Navigation & Orchestration

T Trusted Context & Memory

Daily Meeting Briefing Agent

Read → **Propose** → **Act**

Low autonomy · read-only · illustrative scenario

Job

Before a scheduled meeting, assemble a one-page briefing from authorised internal sources.

Trigger

A meeting is detected for the next working day.

Flow

retrieve relevant records → identify recent changes → assemble briefing → flag gaps → finish.

Always do

Retrieve authorised information; summarise recent changes; separate fact from inference; flag stale or conflicting information.

Ask first

None under normal operation.

Never do

Change source records; send communications; recommend actions outside its mandate.

Failure behaviour

If a source is unavailable or a permission check fails, produce an incomplete briefing with the missing source clearly identified. Never fill the gap by guessing.

Why it matters

An agent can be genuinely agentic without being highly autonomous. It decides what to retrieve and inspect, but its external authority is almost zero.

TODO | **Scenario skeleton. Ahmed to confirm the domain wording before publishing.**

APPENDIX · PROPOSE

Client Follow-Up Agent

Read → **Propose** → Act

Approval-gated action · illustrative scenario

Job

After a meeting, prepare the follow-up, the proposed actions and the record updates.

Trigger

Meeting marked complete.

Flow

meeting notes → extract commitments → check supporting context → draft follow-up → propose CRM and task updates → **ASK FIRST** → execute only approved actions → confirm completion.

Always do

Draft the communication; extract agreed actions and owners; prepare proposed system updates; flag unsupported statements.

Ask first

Send the external communication; write or update CRM records; create consequential follow-up actions.

Never do

Alter approved wording after approval; invent commitments; send when required evidence is missing; treat silence as approval.

Fail safely

If approval expires, the content changes after approval, or the send or update tool fails, stop and report the exact state. Do not blindly retry a consequential action.

Why it matters

The model proposes, the system authorises — and it activates nearly every letter naturally.

TODO | **Scenario skeleton. Ahmed to confirm the domain wording before publishing.**

Operations Exception Resolution Agent

Read → **Propose** → **Act**

Bounded autonomy · illustrative scenario

Job

Monitor incoming service exceptions and resolve standard, reversible cases without human intervention.

Trigger

A new exception enters the queue.

Flow

classify → gather evidence → determine permitted resolution → act → verify result → close or escalate.

Always do

Resolve approved low-risk categories; gather required evidence; make reversible system updates; record sources and actions; verify the result after acting.

Ask first

Anything outside defined thresholds; client-visible changes; cases involving conflicting policies; actions above financial or reputational limits.

Never do

Override access controls; invent a resolution category; perform irreversible actions; continue after repeated failure.

Stop and fail safely

Stop when done, blocked, outside mandate, without progress, or at a ceiling. Preserve current state, explain what was completed, identify the blocked step, and hand off to the named owner.

Why it matters

The agent genuinely chooses and acts, but inside engineered boundaries.

TODO | **Scenario skeleton. Ahmed to confirm the domain wording before publishing.**

Where the old content goes.

For teams already trained on the earlier version of this framework. If you are new to BLUEPRINT, skip this page.

v1.1	v2.0	Where old content goes
Limitations and Non-Goals	Limits & Autonomy	Topic limits stay in BLUE; action columns added
Examples and Test Cases	Examples & Evaluation	Examples stay; evaluation suite added
Parameters and Configuration	Permissions & Identity	Format, length, style → PINPOINT's Narrow the Ask
Requirements	Reliability & Recovery	Accuracy thresholds → E; compliance → L and P
Normalise and Version Control	Navigation & Orchestration	Versioning → the lifecycle loop
Temporal Memory	Trusted Context & Memory	Compaction survives; memory becomes optional